CS351 Project 3 Group 6

Team Member Contributions

Zach Healy

* Email: [zjhealy1s@semo.edu](mailto:zjhealy1s@semo.edu)
* Contribution:
  + Selection Sort Algorithm
  + Encryption Decryption Alogrithms

Jalik Smith

* Email: [jmsmith13s@semo.edu](mailto:jmsmith13s@semo.edu)
* Contribution:
  + Main code for both projects

Momoh Brema

* Email: \_\_\_
* Contribution:
  + Documentation, as well as helping with code when needed

Video Link:

Abstract (150 – 500 words)

Introduction (250 – 500 words)

This is project 2 for group 6, consisting of Zach Healy, Jalik Smith, and Momoh Brema. This project consists of having 2 programs, one that is a selection algorithm which takes a file of numbers, adds them into an array, and then sorts the in numerical order and writes them to a file output. The other program that was written for this project consists of encryption and decryption algorithms. This program will take the input of a file that has a message, then encrypt it to a file output. After that it will take that same file and read it to then be decrypted and have the result message be outputted to a new file. The main goals that were to be focused on for these programs was help with learning when and how to use pointers and arrays in C programming, as well as how to write functions that work within your code. The last thing that we needed to learn for this assignment was to read and write files while using C programming in the Linux Ubuntu environment. We had to learn a lot to do this project to the professional standard that we strive for. For one, we had to learn how to most efficiently use the linux environment. Alongside that, the idea of file input and output in linux was a new concept for us in C. While we have a basis for things like arrays in other languages in Java and Python, every programming language seems to have different ways to read and store files.

Main body (400 – 4000 words)

Discussion (100 – 800 words)

Our results were very pleasing at the end! We ended up getting the programs to work perfectly as we had planned them to, however that was not without its hiccups. The main issues that we faced were with getting the program to write its results to a file for the output. We’ve done this in other languages, but C was just different enough that it gave us quite a bit of trouble. Once we read the textbook and figured out more about the syntax regarding the writing format it made a lot more sense. Along with that, a smaller hiccup we encountered was that our linux environment we were working in crashed and was constantly failing boot in the middle of the project. Luckily we had our code stored somewhere else so we were able to re-install Ubuntu and rewrite the programs in a timely manner.

Conclusion (100 – 800 words)

References

??

Appendix

Selection\_sort.c

#include<stdio.h>

//Define function

double\* selection\_sort(double a[]);

int main(){

//define variables

int i = 0;

double num;

double \*p;

double arr[36];

//open file and assign each number to point in array

FILE \*file = fopen("num.txt", "r");

while(fscanf(file, "%le, ", &num) > 0){

arr[i] = num;

i++;

}

//run function to sort array

p = selection\_sort(arr);

//load file to store array to, then store array to file

FILE \*aNum = fopen("ascended\_num.txt", "wb");

for(int i = 0; i < 36; i++){

fprintf(aNum, "%f\n", p[i]);

}

fclose(aNum);

}

//function for selection sort

double\* selection\_sort(double arr[]){

//define variables

int temp = 0;

double place = 0;

//selection sort alogrithm

for(int i = 0; i < 36; i++){

temp = i;

for(int j = i+1; j < 36; j++){

if(arr[j] < arr[temp]){

temp = j;

}

}

//after comparison, move array elements

place = arr[temp];

arr[temp] = arr[i];

arr[i] = place;

}

//return sorted array

return arr;

}

encryptDecrypt.c

#include<stdio.h>

//define functions that will be used

char\* encrypt(char arr[]);

char\* decrypt(char arr[]);

//define size that will be used for array

//and array scanning

#define SIZE 74

int main(){

//define variables used

char arr[SIZE];

char temp;

int i;

//Read message.txt and assign it to variable

FILE \*file = fopen("message.txt", "r");

while(fscanf(file, "%c", &temp) > 0){

arr[i] = temp;

i++;

}

//encryption and print to file.

encrypt(arr);

FILE \*encrypt = fopen("encrypted\_message.txt", "w");

fputs(arr, encrypt);

//for(int i = 0; i < SIZE; i++){

// fprintf(encrypt, "%c", arr[i]);

//}

//fclose(encrypt);

//read encrypted file and assign to array

FILE \*encrypted = fopen("encrypted\_message.txt", "r");

while(fscanf(encrypted, "%c", &temp) > 0){

arr[i] = temp;

i++;

}

//decrypt and print to file

decrypt(arr);

FILE \*result = fopen("result\_message.txt", "w");

fputs(arr, result);

//for(int i = 0; i < SIZE; i++){

// fprintf(result, "%c", arr[i]);

//}

//fclose(result);

}

//encryption function (shift cipher)

char\* encrypt(char arr[]){

for(int i = 0; i < SIZE; i++){

arr[i] -= 3;

}

}

//decryption function (shift cipher)

char\* decrypt(char arr[]){

for(int i = 0; i < SIZE; i++){

arr[i] += 3;

}

}